Docker: Beyond the Basics

CI/CD (Day Two)

Release 2021-05-29-1040

Instructor

Sean P. Kane

@spkane

@superorbital_io

SUPERORBITAL

Follow Along Guide

Textual Slides

Relaunch Services

```
$ cd ~/class-docker-cicd/layout
$ cd compose/final
$ docker-compose up -d
```

Getting Started with Jenkins

- Navigate web browser to:
  - http://127.0.0.1:10081/
- Login to Jenkins
  Click: `create new jobs`

Note: If you have not configured Jenkins, you can login using the `admin` user and the `initialAdminPassword`.

Creating The Jenkins Job

Enter an item name: `outyet`
Click: `Freestyle project`
Click: `OK`

Configuring the Job

Description: `build and test outyet`

Gogs Webhook

- Gogs Secret: `12345`

Source Code Management

Select: `git`
Repository URL: http://gogs:3000/myuser/outyet.git
Branch Specifier (blank for 'any'): ``

Build Triggers

- None

Build Environment

Check: `Delete workspace before build starts`
Check: `Mask passwords and regexes`
Name/Password Pairs:

- **Name**: `DOCKER_PW`
- **Password**: `myuser-pw!`

The Build Script

- Select 'Execute Shell'

```
# This is not ideal, but reasonable for class.
echo "${DOCKER_PW}" | docker login --username=myuser \
    --password-stdin 127.0.0.1:5000
docker image build -t 127.0.0.1:5000/myuser/outyet:${GIT_COMMIT} .
docker image push 127.0.0.1:5000/myuser/outyet:${GIT_COMMIT}
```

Post-Build Actions

- None

Click: Save

Build The Code

Click: `Build Now`

Build The Code

Click: `#1`
Click: `Console Output`

Build Results

Looking for:

```
Finished: SUCCESS
```

Automate Builds

- Navigate web browser to:
  - http://127.0.0.1:10080/
    Click: `outyet`
    Click: `Settings`
    Click: `Webhooks`

Automate Builds

Click: `Add Webhook`
Click: `Gogs`
Payload URL:
`http://jenkins:8080/gogs-webhook/?job=outyet`
Content Type: `application/json`
Gogs Secret: `12345`
When should this web hook be triggered?: `Just the push event`
Check: `Active`
Click: `Add Webhook`

Automate Builds

Click: `http://jenkins:8080/gogs-webhook/?job=outyet`
Click: `Test Delivery`

- Confirm Green Checkmark

Add a Bug

```
$ cd ~/class-docker-cicd/code/outyet
$ vi main.go
```

- Modify to look like this:
  - Add `"net/url"`

Result

```
import (
    "expvar"
    "flag"
    "fmt"
    "html/template"
    "log"
    "net/http"
    "net/url"
    "os"
    "sync"
    "time"
)
```

Commit the Bug

```
$ git add .
$ git commit -m "Introducing bug"
$ git push origin master
```

Failed Tests

```
./main.go:28: imported and not used: "net/url"
[0mThe command '/bin/sh -c \
  go get -v -d && go install -v && go test -v && \
  go build -ldflags "-s" -a -installsuffix cgo -o outyet .' \
  returned a non-zero code: 2
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

Fix The Error

```
$ cd ~/ class-docker-cicd/code/outyet
$ vi main.go
```

- Modify to look like this:
  - Remove `"net/url"`

Result

```
import (
    "expvar"
    "flag"
    "fmt"
    "html/template"
    "log"
    "net/http"
    "os"
    "sync"
    "time"
)
```

Commit the Fix

```
$ git add .
$ git commit -m "Removing bug"
$ git push origin master
```

Successful Tests

```
deploy_e0ebf86decf4795cee332523e68017ce7952e094:
  digest:
  sha256:a31bc49ececbac7d79b1dd080b5167ee55b34c385e967e48bfd107f8ba5afbee
  size: 738
Finished: SUCCESS
```

Tear Down Pipeline

```
$ docker-compose down
```

Orchestration Tools

- Kubernetes
  - https://kubernetes.io/
- Docker swarm-mode
  - https://docs.docker.com/engine/swarm/
- DC/OS Community Edition (Mesos/Marathon)
  - https://dcos.io/

Terraform

- https://www.terraform.io/

```
$ cd ~/class-docker-cicd/layout/terraform
$ terraform init
$ terraform plan
$ terraform apply
Apply complete! Resources: 4 added, 0 changed, 4 destroyed.
Outputs:
...
```

```
$ source ./bin/ip_vars.sh
```

Primary Swarm Manager

```
$ ssh -i ~/.ssh/${KEY} ubuntu@${MASTER_IP}
```

```
$ sudo docker swarm init
$ sudo docker swarm join-token --quiet worker
$ sudo docker swarm join-token --quiet manager
$ exit
```

Secondary Swarm Managers

```
$ export SWARM_MANAGER_TOKEN=${MANAGER_TOKEN}
$ ./bin/setup_managers.sh
$ docker -H ${primary_manager_ip}:2375 node list
```

- We are running this command on each node:
  - ```
    sudo docker swarm join --token ${SWARM_MANAGER_TOKEN}
    ${MASTER_IP}:2377
    ```
    ```

Swarm Workers

```
$ export SWARM_WORKER_TOKEN=${WORKER_TOKEN}
$ ./bin/setup_workers.sh
$ docker -H ${primary_manager_ip}:2375 node list
```

- We are running this command on each node:
  - ```
    sudo docker swarm join --token ${SWARM_WORKER_TOKEN}
    ${MASTER_IP}:2377
    ```
    ```

Create Network

```
$ docker -H ${primary_manager_ip}:2375 network create \
    --driver=overlay my-net
```

Create a Service

```
$ docker -H ${primary_manager_ip}:2375 service create \
    --detach=true --name outyet0 --replicas 4 \
    --publish published=80,target=8080 \
    --network my-net spkane/outyet:1.9.3
$ curl http://${primary_manager_ip}:80
```

Examining the Service

```
$ docker -H ${primary_manager_ip}:2375 service ls
$ docker -H ${primary_manager_ip}:2375 service inspect --pretty outyet0
$ docker -H ${primary_manager_ip}:2375 service ps outyet0
```

Scaling the Service

```
$ docker -H ${primary_manager_ip}:2375 service scale \
    --detach=false outyet0=8
$ docker -H ${primary_manager_ip}:2375 service ps outyet0
$ docker -H ${primary_manager_ip}:2375 service scale \
    --detach=false outyet0=4
```

Self Healing

```
$ ./bin/self_healing.sh
```

Update the Service

```
$ docker -H ${primary_manager_ip}:2375 service update \
    --update-delay 10s --update-failure-action rollback \
    --update-monitor 2s --update-order start-first \
    --update-parallelism 1 --detach=false \
    --image spkane/outyet:1.9.4 outyet0
$ curl http://${primary_manager_ip}:80
```

Rollback the Service

```
$ docker -H ${primary_manager_ip}:2375 service rollback outyet0
$ curl http://${primary_manager_ip}:80
```

Build & Deploy Script

- ${MANAGER_IP} and ${HUB_USER} must be replaced with real values in the script below.

```
echo "${DOCKER_PW}" | docker login --username=myuser \
    --password-stdin 127.0.0.1:5000
docker image build -t 127.0.0.1:5000/myuser/outyet:${GIT_COMMIT} .
docker image push 127.0.0.1:5000/myuser/outyet:${GIT_COMMIT}
docker -H ${MANAGER_IP}:2375 service update \
  --update-delay 10s --update-failure-action rollback \
  --update-monitor 2s --update-order start-first \
  --update-parallelism 1 --detach=false \
  --image 127.0.0.1:5000/myuser/outyet:${GIT_COMMIT} outyet0
```

Drain a Node

```
$ docker -H ${primary_manager_ip}:2375 service ps outyet0
$ docker -H ${primary_manager_ip}:2375 node ls
$ docker -H ${primary_manager_ip}:2375 node update \
    --availability drain ${NODE_NAME}
$ docker -H ${primary_manager_ip}:2375 node inspect --pretty ${NODE_NAME}
$ docker -H ${primary_manager_ip}:2375 service ps \
  -f "desired-state=running" outyet0
$ docker -H ${primary_manager_ip}:2375 node update \
    --availability active ${NODE_NAME}
```

Deleting the Services

```
$ docker -H ${primary_manager_ip}:2375 service rm outyet0
$ docker -H ${primary_manager_ip}:2375 service inspect outyet0
$ docker -H ${primary_manager_ip}:2375 network rm my-net
```

Destroying Infrastructure

```
$ cd ~/class-docker-cicd/layout/terraform
terraform destroy
Destroy complete! Resources: # destroyed.
```

What We Have Learned

- Configuring Jenkins
- Building & Testing with Jenkins
- Automating builds with web hooks
- Overview of orchestration tooling
- Building a Docker swarm-mode cluster
- Creating and managing scalable services

Additional Reading

- The 12-Factor App
  - http://12factor.net/
- Official Docker Documentation
  - https://docs.docker.com/
- Docker: Up and Running
  - http://shop.oreilly.com/product/0636920153566.do

Additional Learning Resources

https://learning.oreilly.com/

Student Survey

Please take a moment to fill out the class survey linked to from the bottom of the ON24 audience screen.

O'Reilly and I value your comments about the class.

Thank you!

Any Questions?

Sean P. Kane

Providing stellar Kubernetes engineering and workshops.

https://superorbital.io/contact/